APPLICATION

FOR

UNITED STATES LETTERS PATENT

APPLICANT NAME: Bruce Wallman

TITLE: SYSTEM AND METHOD FOR ELIMINATING VIRUSES AT A WEB PAGE SERVER

DOCKET NO.: CHA920030013US1

INTERNATIONAL BUSINESS MACHINES CORPORATION

CERTIFICATE OF MAILING UNDER 37 CFR 1.10

I hereby certify that, on the date shown below, this correspondence is being deposited with the United States Postal Service in an envelope addressed to: Mail Stop Patent Application, Commissioner for Patents, P.O. Box 1450, Alexandria VA 22313-1450 as "Express Mail Post Office to Addressee" Mailing Label No. EV108090027US

on July 29, 2003

Wendy E. Thompson
Name of person mailing paper

Wendy E. Thompson 7/29/2003
Signature Date

SYSTEM AND METHOD FOR ELIMINATING VIRUSES AT A WEB PAGE SERVER

BACKGROUND OF THE INVENTION

1. Technical Field

[001] The present invention relates generally to anti-virus systems, and more specifically relates to a system and method that avoids storing and forwarding web pages that can carry viruses.

2. Related Art

[002] Viruses are prevalent today throughout the Internet, and can be spread in many different ways. One technique for spreading viruses is to embed a virus in a web page as an active web page element. Then, when a user loads the web page from a web server, the virus can be launched onto the user's computer. Often, a web page server receives and stores web pages from many different sources and/or authors. Accordingly, identifying these types of viruses can be a difficult and time-consuming process for the web server.

[003] One way to address this problem is to run virus-checking programs on the web server to scan for viruses and remove any residual viruses. Unfortunately, virus-checking programs provide several drawbacks including: (1) they consume computational resources; (2) that must be run often to be effective; and (3) they are limited by the effectiveness of the scanning program.

[004] For instance, U.S. Patent Application US 2002/0103783 A1, by Muhlestein, "Decentralized Virus Scanning For Stored Data," filed on 8/1/2002, which is hereby incorporated by reference, provides a system for scanning requested files for viruses. The invention, however, requires an external computing device that opens requested files and scans for viruses any time there appears to be some risk. Thus, the system potentially consumes a significant amount of time and resources.

[005] Accordingly, a need exists for a web server system that can eliminate viruses without incurring such computational overhead.

SUMMARY OF THE INVENTION

[006] The present invention addresses the above-mentioned problems, as well as others, by providing a system and method for eliminating viruses at a web server by storing pages without any active web page elements. In a first aspect, the invention provides a web server having anti-virus protection, comprising: an active element filter for stripping active elements from web pages being stored at the web server; and an active element insertion system for inserting active elements into stored web pages when the web page is requested.

[007] In a second aspect, the invention comprises a method for providing anti-virus protection to a web server, comprising: receiving web pages that are to be stored at the web server; stripping active elements from the web pages being stored at the web server; storing the web pages at the web server; receiving a request for a web page to be served by the web server; determining if active elements are required for the requested web

page; inserting active elements into the requested web page if active elements are required; and serving the requested web page.

[008] In a third aspect, the invention provides a program product stored on recordable medium for providing anti-virus protection to a server, comprising: means for stripping active elements from files being stored at the server; means for determining if active elements are required for a file being requested from the server; and means for inserting an active element into the requested file if an active element is required.

BRIEF DESCRIPTION OF THE DRAWINGS

[009] These and other features of this invention will be more readily understood from the following detailed description of the various aspects of the invention taken in conjunction with the accompanying drawings in which:

[010] Figure 1 depicts a web server having an anti-virus system in accordance with the present invention.

[011] Figure 2 depicts a flow diagram of a method of implementing the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[012] Referring now to the drawings, Figure 1 depicts a web server 10 having a system for eliminating viruses stored in active elements of web pages. In particular, web server 10 includes an active element filter 14 that strips active elements from web pages 22 being stored at the web server 10. By doing this, web server 10 eliminates the

dissemination of viruses launched from active elements in a web page 24 being served by web server 10.

[013] An active element may include any type of code fragment that instructs a browser, client, interpreter, virtual machine, processor, etc., to perform some action when the code fragment is encountered. In the exemplary embodiments described herein, active elements are described with reference to JavaScript constructs stored in web pages. However, it should be understood that the invention could apply to any type of active element (e.g., VBScript, Tcl, Perl, Rexx, etc.) stored in any type of servable file. Exemplary uses for JavaScript constructs in a Web environment includes performing actions such as: (1) automatically changing a formatted date on a Web page; (2) causing a linked-to page to appear in a popup window; and (3) causing text or a graphic image to change during a mouse rollover.

[014] The filtering process may strip out active elements from web pages in any manner, e.g., with a software program that identifies and removes JavaScript constructs. In a typical browser application, the number of different types of JavaScript constructs is relatively small, e.g., 6-12. Accordingly, identifying and eliminating the constructs is a relatively trivial task. Once filtered, the stripped web page 23 is stored in web page storage 26. If no active elements exist in a web page 22 that is to be stored, then no filtering is required, and the web page 22 is simply stored in web page storage 16. If active elements exist that are needed when the page is stored, these can be noted by XML or similar replacement elements.

[015] When the web server 10 receives a request 26 for a web page 23, e.g., from a client or other server, retrieving system 18 fetches the web page 23 from web page

storage 16. Retrieving system 18 then determines if the web page 23 requires any active elements. This determination can be made by looking for predetermined additions to each page, or by finding stored requests, such as the XML elements noted above, and replacing these elements with programmed active elements known to be 'safe.' If no active elements are required, then the web page is served. If one or more active elements are required, then they are added in by active element insertion system 20.

[016] Active elements can be added to the web page 23 in any manner. In one exemplary embodiment, the active elements could be stored in compiled server code 28. This implementation has the advantage of providing heightened security since no one can construct and leave active files with active elements on the server other than the authors of the server themselves. As an alternative embodiment, the active elements can be stored in files 30 separate from the web page storage 16, and accessed by active element insertion system 20. As noted above, the number of different type of active elements in a browser application is relatively small. Accordingly, implementing the active element insertion system 20 is a fairly simple operation for one skilled in the art.

[017] In order to insert active elements back into web page 23, active element insertion system 20 will typically need to first examine the attributes of web page 23. For example, suppose that it is desired to bring every page of a certain type to a front-most window on the client's computer. A bring-to-front JavaScript element could be added to every page being sent to clients by adding it as an 'onLoad' request at the start of every HTML page using the active element insertion system 20. If this type of element is needed for only certain pages, this could be identified either by something unique to these pages such as something special about their names, or it could be determined by placing

XML code such as <newElement>Bring-To-Front</newElement> in the proper pages and then replacing this with 'safe' active code using the active element insertion system 20. Once the active elements are inserted, the web page 24 containing the active elements is served to the requesting entity. Accordingly, it is impossible to infect the web server 10 with a virus stored in an active element of a web page.

[018] Referring now to Figure 2, a flow diagram of a method of implementing the invention is shown. At step S1, the server receives a web page to be stored. At step S2, the server filters out any active elements from the web page, and stores the web page at step S3. At step S4, the server receives a request for a web page, and fetches the web page from storage at step S5. Next, at step S6, a determination is made whether there are active elements required for the requested page. If no active elements are required, the web page is served at step S8. Otherwise, if active elements are required, then the server adds back the active elements at step S7, and serves the web page with the active elements inserted at step S8.

[019] It is understood that the systems, functions, mechanisms, methods, and modules described herein can be implemented in hardware, software, or a combination of hardware and software. They may be implemented by any type of computer system or other apparatus adapted for carrying out the methods described herein. A typical combination of hardware and software could be a general-purpose computer system with a computer program that, when loaded and executed, controls the computer system such that it carries out the methods described herein. Alternatively, a specific use computer, containing specialized hardware for carrying out one or more of the functional tasks of the invention could be utilized. The present invention can also be embedded in a

computer program product, which comprises all the features enabling the implementation of the methods and functions described herein, and which - when loaded in a computer system - is able to carry out these methods and functions. Computer program, software program, program, program product, or software, in the present context mean any expression, in any language, code or notation, of a set of instructions intended to cause a system having an information processing capability to perform a particular function either directly or after either or both of the following: (a) conversion to another language, code or notation; and/or (b) reproduction in a different material form.

[020] The foregoing description of the preferred embodiments of the invention has been presented for purposes of illustration and description. They are not intended to be exhaustive or to limit the invention to the precise form disclosed, and obviously many modifications and variations are possible in light of the above teachings. Such modifications and variations that are apparent to a person skilled in the art are intended to be included within the scope of this invention as defined by the accompanying claims.